

IN THE CLAIMS

Please amend the claims as follows:

1. (Original) A database management system, comprising:

a processor associating lock durations with different activities in a transaction, and maintaining locks for the duration of the activities and then releasing the locks when the activities are completed.

2. (Currently Amended) A database management system according to claim 1 wherein one of the activities ~~includes a operation~~ include a group of individual sort operations and the processor activates locks for ~~each separate sort operation~~ each of the individual sort operations in the group and releases the locks only when the all of the individual sort operations in the group are completed.

3. (Original) A database management system according to claim 1 including a memory containing an activity bit map that tracks individual activities for the transaction, the processor assigning activity identifiers to the activities according to the activity bit map.

4. (Original) A database management system according to claim 1 wherein the processor associates the activities with lock modes and releases the lock modes on data items when the associated activities are finished.

5. (Original) A method for locking data items in a database management system, comprising:

associating lock durations with different activities in a transaction;
maintaining locks for the duration of the activities; and
releasing the locks when the activities are completed.

6. (Currently Amended) A method according to claim 5 including:

identifying a plurality of different sort activities operations for the same transaction;
activating locks on data items associated with the plurality of different sort activities operations; and

releasing the locks on the data items only when all of the associated plurality of different sort activities operations are completed.

7. (Original) A method according to claim 5 including:

maintaining an activity bit map that tracks individual activities for the transaction; and
assigning activity identifiers to the activities according to the activity bit map.

8. (Currently Amended) A method according to claim ~~4~~ 6 including associating the activities with lock modes and releasing the lock modes on data items when the associated activities are finished.

a processor associating lock durations with different activities in a transaction, and maintaining locks for the duration of the activities and then releasing the locks when the activities are completed.

9. (New) A method according to claim 1 including:

assigning a same unique activity identifier to multiple different arbitrary database access instructions that constitute the different activities in the transaction, the database access instructions performing one or more operations on multiple data items in a database and the activity identifier assigned to and associated with the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions;

assigning multiple locks to the multiple data items corresponding with the operations performed on the multiple data items pursuant to the database access instructions; and

preventing other transactions and other associated activities from accessing the multiple data items until all of the multiple operations are completed for all of the database access instructions assigned to the activity identifier.

10. (New) A database management system, comprising:

a processor configured to assign activity identifiers to database access instructions that perform one or more operations on multiple data items in a database, the activity identifiers assigned to and associated with the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions,

the processor further configured to assign multiple locks to the multiple data items corresponding with the operations performed on the multiple data items pursuant to the database access instructions and further configured to only release the multiple locks when all of the multiple operations are completed for all of the database access instructions assigned to the same activity identifiers.

11. (New) The database management system according to claim 10 wherein the processor is further configured to assign the activity identifiers to an arbitrary group of related database access instructions performing operations on an arbitrarily related group of data items.

12. (New) The database management system according to claim 10 wherein the processor is further configured to assign common transaction identifiers to different related groups of database access instructions assigned different activity identifiers and coordinate when the different related groups of database access instructions are allowed to perform operations on the data items.

13. (New) A database management system according to claim 10 wherein the processor is configured to assign a first transaction identifier to a group of individual sort operations and assign locks to the data items associated with the sort operations, the processor further configured to hold the locks until all of the individual sort operations in the group have been completed.

14. (New) Computer readable media containing instructions that when executed by a computer, comprise:

assigning activity identifiers to database access instructions that perform multiple operations on multiple data items in a database, the activity identifiers assignable to the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions; and

assigning multiple locks to the multiple data items corresponding with the operations performed on the multiple data items by the database access instructions and only releasing the multiple locks when all of the multiple operations are completed for all of the database access instructions assigned to the same activity identifiers.

15. (New) The computer readable media according to claim 14 including instructions that when executed assign the activity identifiers to an arbitrary group of related database access instructions performing operations on an arbitrarily related group of data items.

16. (New) The computer readable media according to claim 14 including instructions that when executed assign common transaction identifiers to different related groups of database access instructions each assigned different activity identifiers and coordinate when the related groups of database access instructions are allowed to perform operations on the data items.

17. (New) The computer readable media according to claim 14 including instructions that when executed assign a first transaction identifier to a group of individual sort operations, assign

locks to the data items associated with the sort operations, and to hold the locks until all of the individual sort operations in the group have been completed.

18. (New) A database management system, comprising:

means for associating lock durations with different activities in a transaction;

means for maintaining locks for the duration of the activities; and

means for releasing the locks when the activities are completed.

19. (New) The database management system according to claim 18 including:

means for identifying a plurality of different sort activities for the same transaction;

means for activating locks on data items associated with the sort activities; and

means for releasing the locks on the data items when the associated sort activities are completed.

20. (New) The database management system according to claim 18 including:

means for maintaining an activity bit map that tracks individual activities for the transaction associated with a same transaction; and

means for assigning activity identifiers to the activities according to the activity bit map.

21. (New) The database management system according to claim 18 including means for identifying one or more subclasses of activities within an activity and associating lock durations with the subclass and releasing the locks upon completion of the activities in the subclass before releasing the locks on the activity.